

# [11g RAC 설치] RAC2 VM 구성

소유자	쏘니
태그	RAC
다중 선택	

목차

1. 개요
2. 리눅스 (OEL 6) 설치
3. OS 환경 설정
4. 공유 스토리지 설정
5. **RAC2 VM 구성**
6. GRID 설치
7. ASM Disk Group 생성
8. DBMS 설치
9. DataBase 생성

---

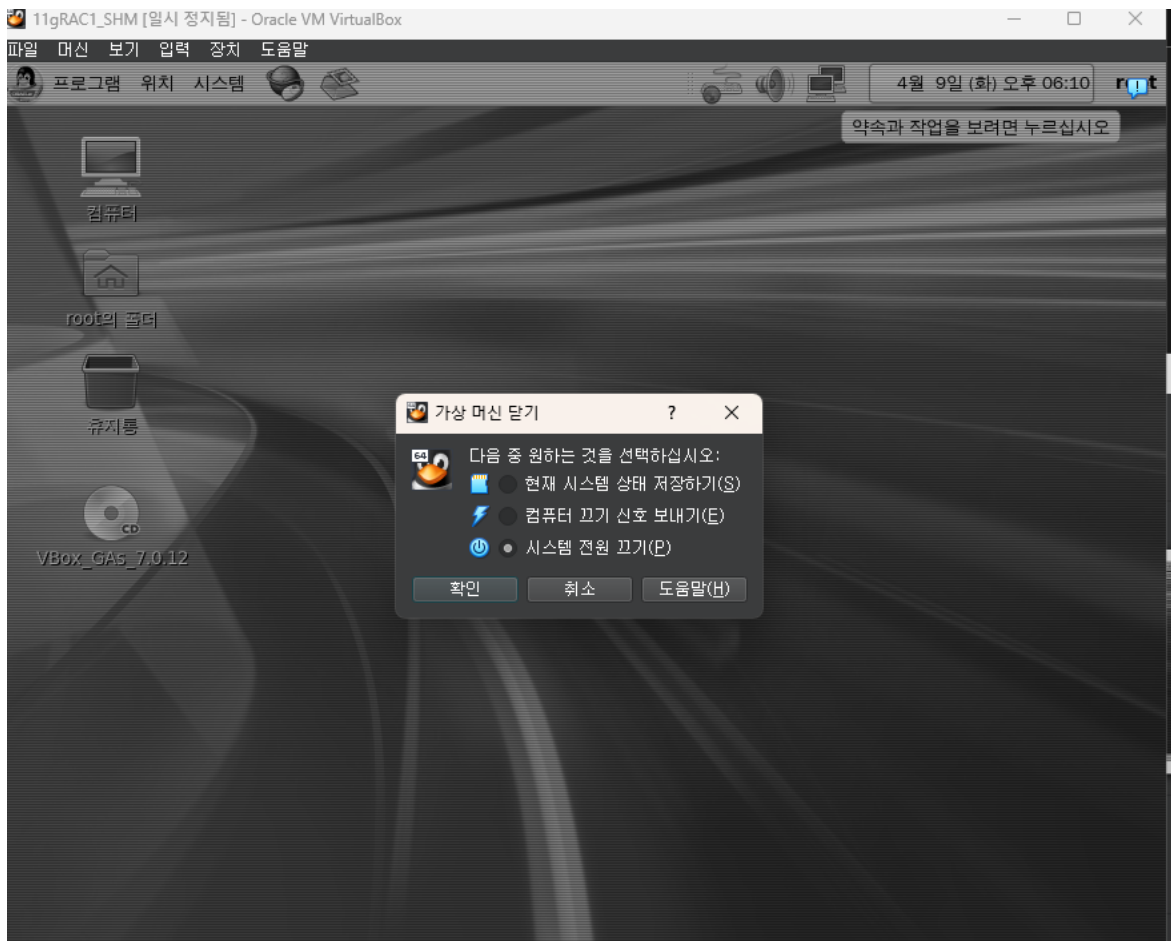
## V. RAC2 VM 구성

### RAC2 VM(2번 노드) 설정하기

---

### [ 복제본 생성(VM 복제) ]

1. 1번 노드를 정상적으로 shutdown 시킵니다. (시스템 전원 끄기)

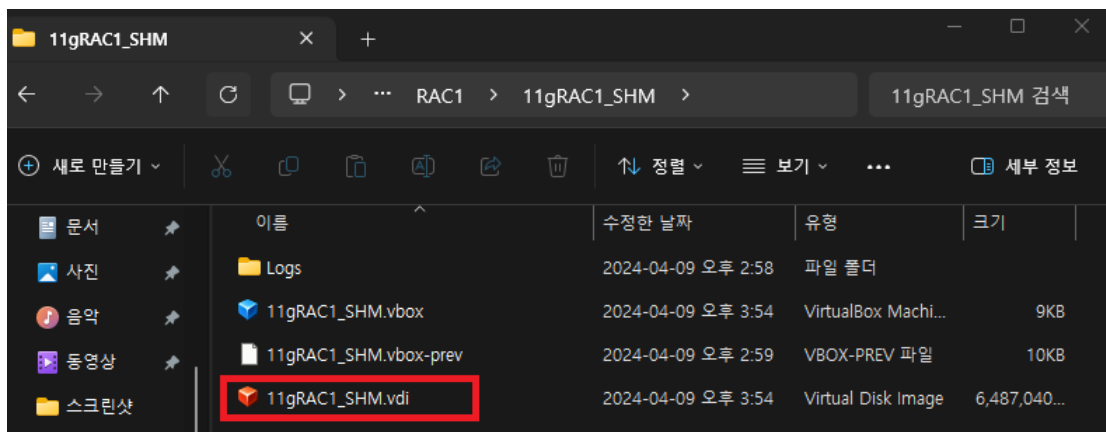


- 윈도우에서 명령프롬프트 창을 관리자 권한으로 엽니다.  
rac1.vdi 가 있는곳으로 이동해서 아래의 명령어를 실행합니다.

\* 자신의 \*.vdi 가 들어있는 디렉토리로 이동하고, \*.vdi 에 맞게 바꿔서 실행하세요!

```
vboxmanage clonemedium <원본>.vdi <복제본>.vdi --format VDI
```

▼ 수행해보기



```

C:\Windows\System32>cd ..

C:\Windows>cd ..

C:\>cd RAC1

C:\RAC1>cd 11gRAC1_SHM

C:\RAC1\11gRAC1_SHM>SET PATH=%PATH%;C:\Program Files\Oracle\VirtualBox

C:\RAC1\11gRAC1_SHM>vboxmanage clonemedium 11gRAC1_SHM.vdi ..\11gRAC2_SHM.vdi --
format VDI

0%...10%...20%...30%...40%...50%...60%...70%...80%...90%...100%
Clone medium created in format 'VDI'. UUID: da7450c7-72cf-43e2-aa71-28cb9ec0c536

```

```

C:\Windows\System32>cd ..

C:\Windows>cd ..

C:\>cd RAC1

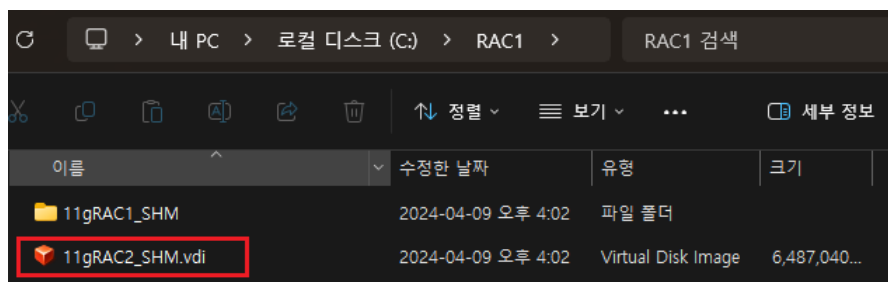
C:\RAC1>cd 11gRAC1_SHM

C:\RAC1\11gRAC1_SHM>SET PATH=%PATH%;C:\Program Files\Oracle\VirtualBox

C:\RAC1\11gRAC1_SHM>vboxmanage clonemedium 11gRAC1_SHM.vdi ..\11gRAC2_SHM.vdi --fo
rmat VDI
0%...10%...20%...30%...40%...50%...60%...70%...80%...90%...100%
Clone medium created in format 'VDI'. UUID: da7450c7-72cf-43e2-aa71-28cb9ec0c536
C:\RAC1\11gRAC1_SHM>

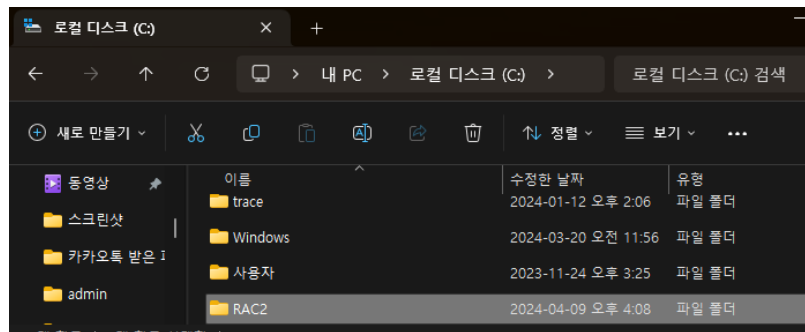
```

3. 다 되었으면 rac2.vdi 가 생성되었는지 확인합니다. ( 본인의 경우 11gRAC2\_SHM.vdi)



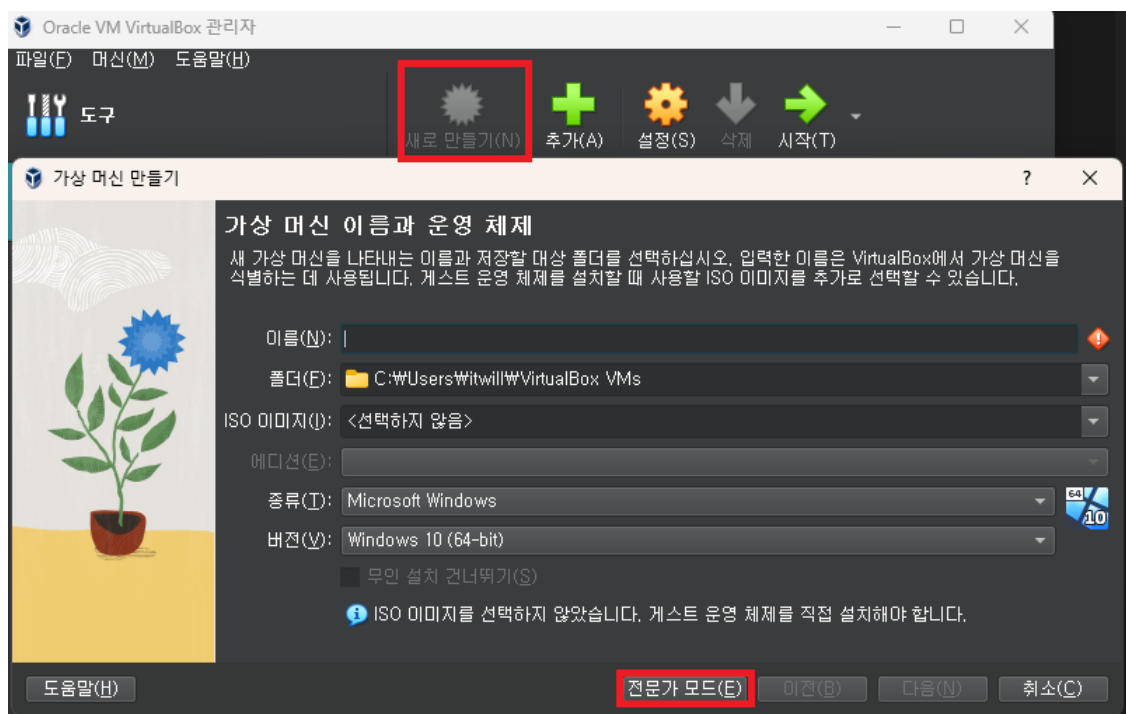
## [ 2번 노드 VM 생성 및 환경 설정 ]

1. C 혹은 D드라이브에 RAC2 폴더 생성



## 2. 가상 머신 만들기

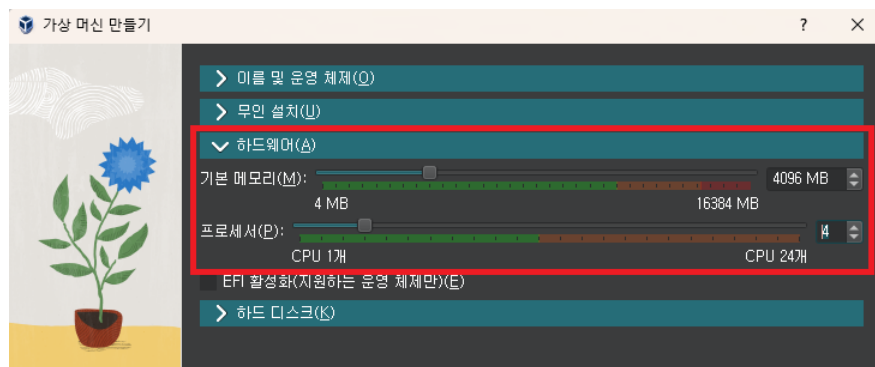
a. VirtualBox에서 '새로 만들기'를 클릭 > 전문가 모드 선택



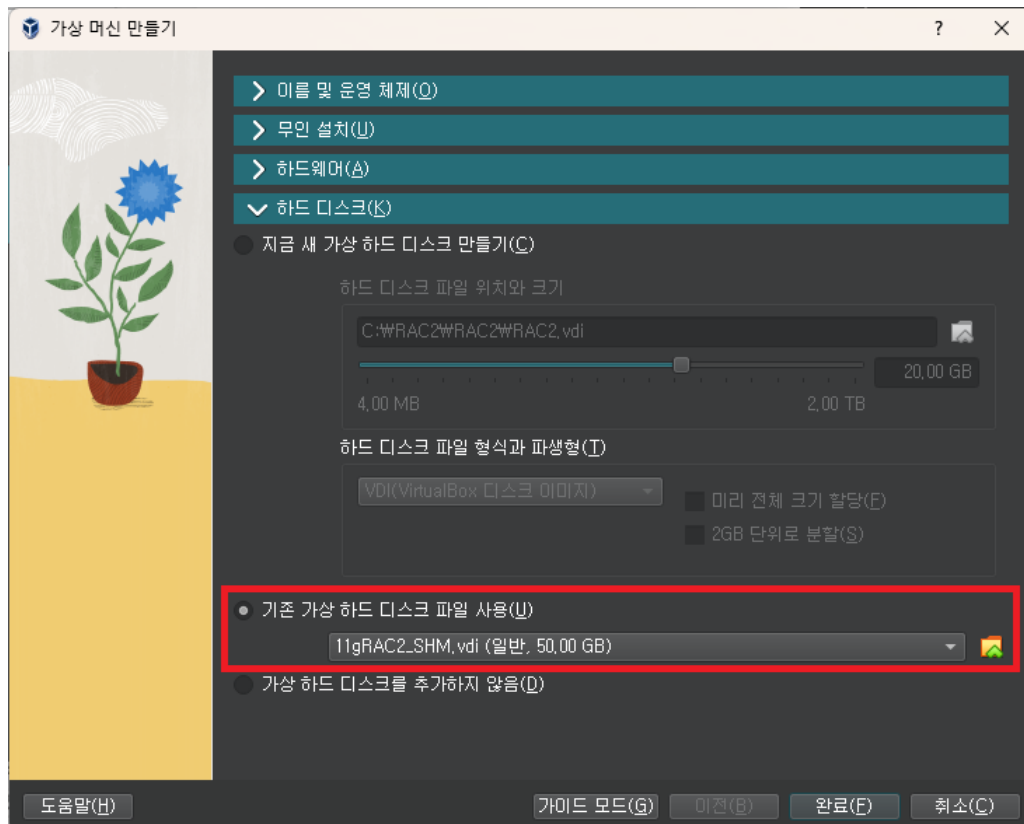
b. 이름 및 운영체제 : 아래와 같이 구성합니다.



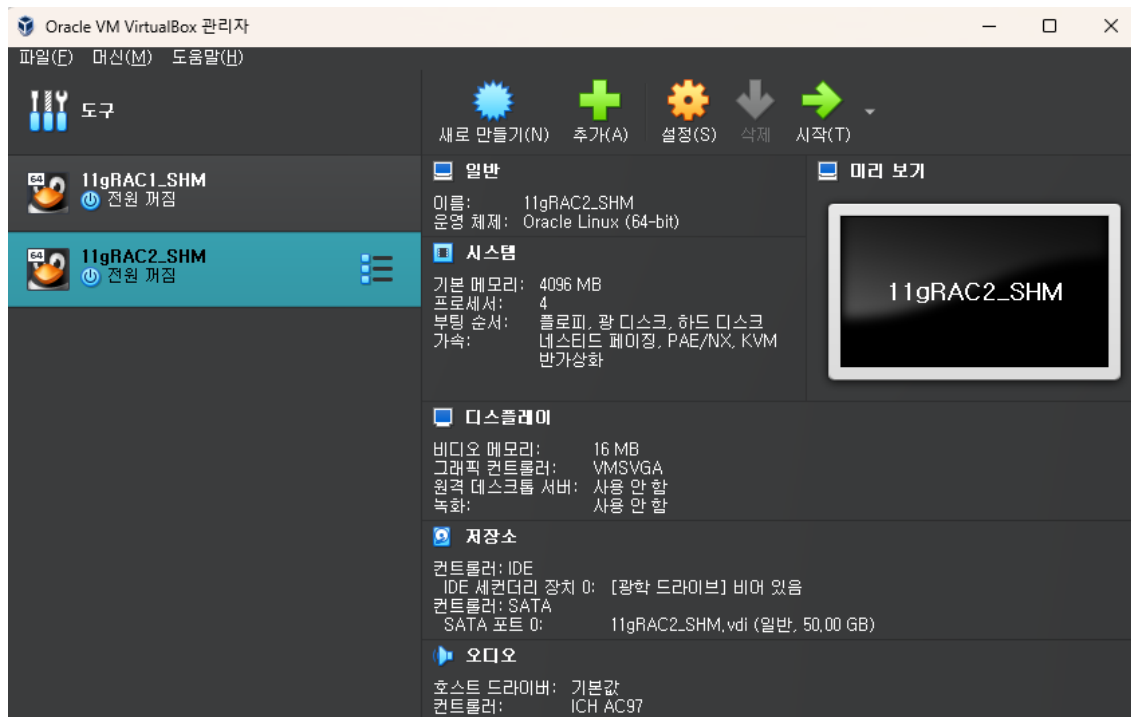
c. 하드웨어 : 기본 메모리 4기가, 프로세서 4개로 변경합니다.



d. 하드디스크 : 기존 가상 파일 하드디스크 사용을 클릭하고 rac2.vdi(본인 11gRAC2\_SHM)를 선택합니다.



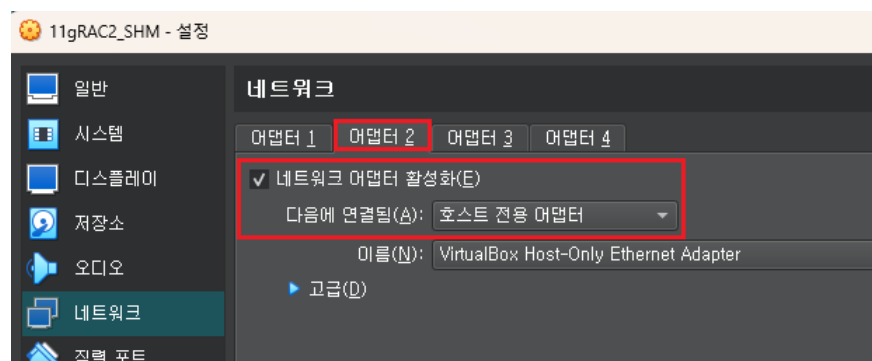
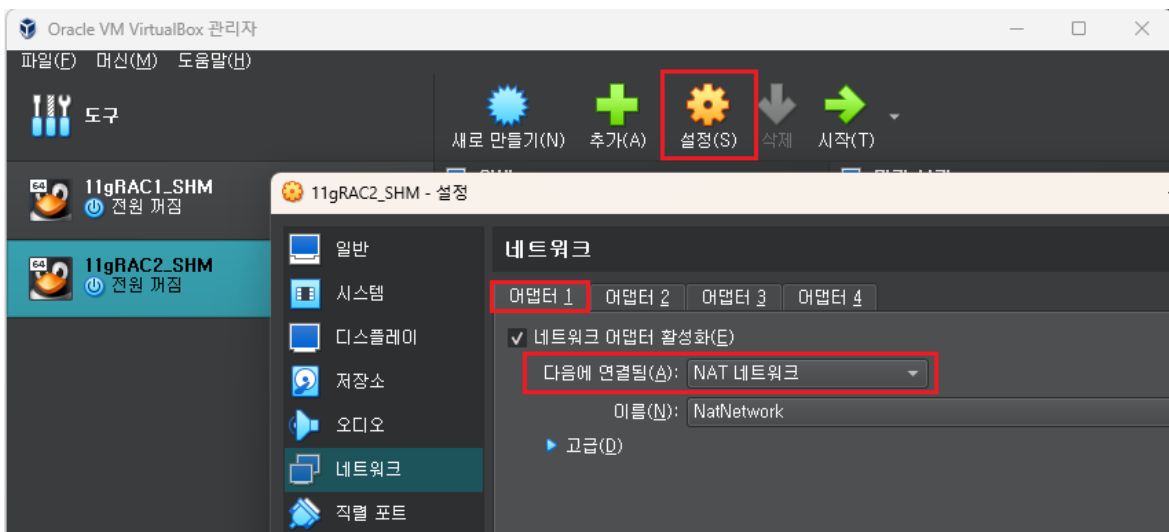
e. 그림 아래와 같이 RAC1과 RAC2가 생겨납니다.



3. 네트워크 구성을 수정할 것이므로 해당 VM을 선택하고 <설정>을 클릭합니다.

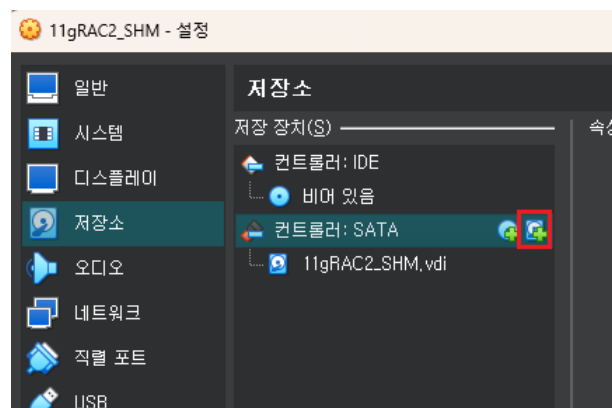
a. 어댑터 1: NAT 네트워크로 설정

b. 어댑터 2 : 호스트 전용 어댑터로 설정

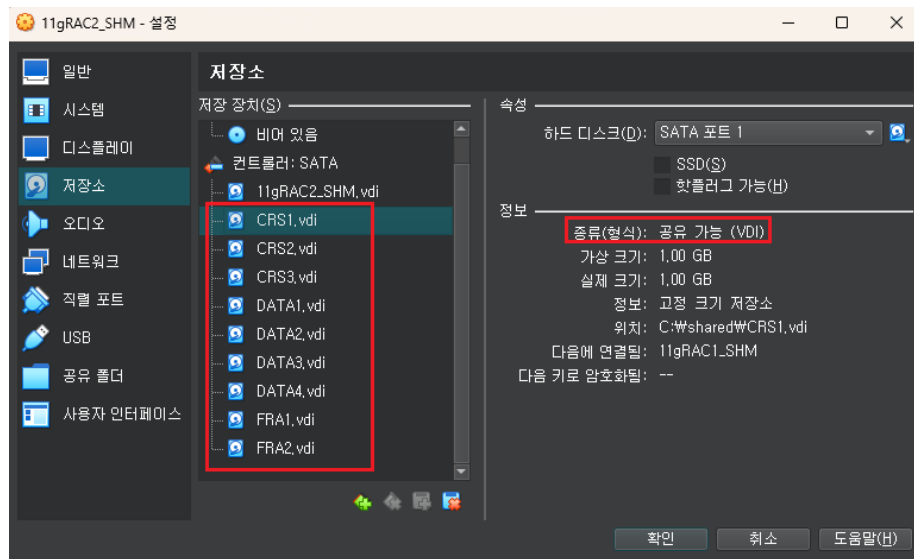


4. 저장소를 구성합니다.

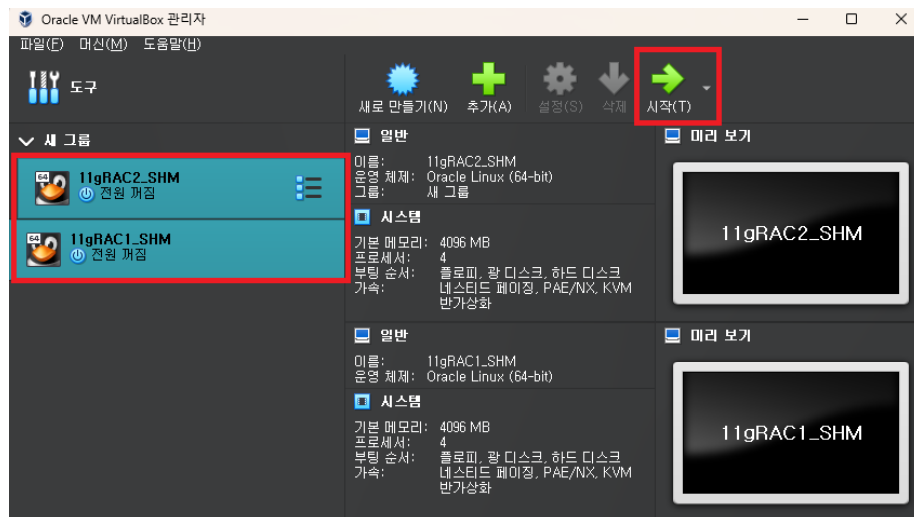
a. 첫번째 노드와 똑같은 순서로 모든 공유 디스크를 연결합니다.



그리고 모두 공유가능한지 확인합니다.



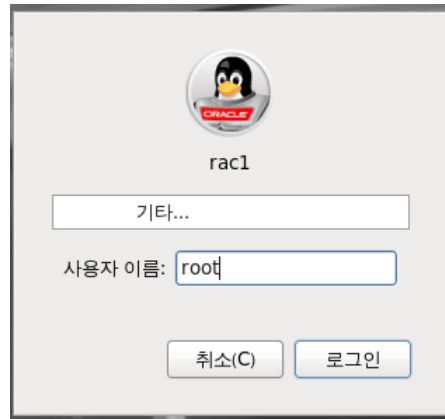
5. 1번 노드와 2번 노드를 Ctrl 키와 함께 선택한 후 > 시작 을 눌러 동시에 시작합니다.



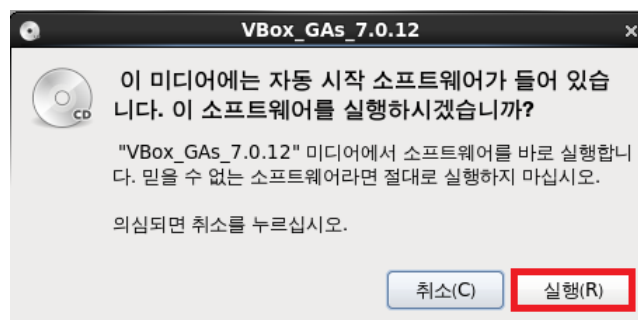
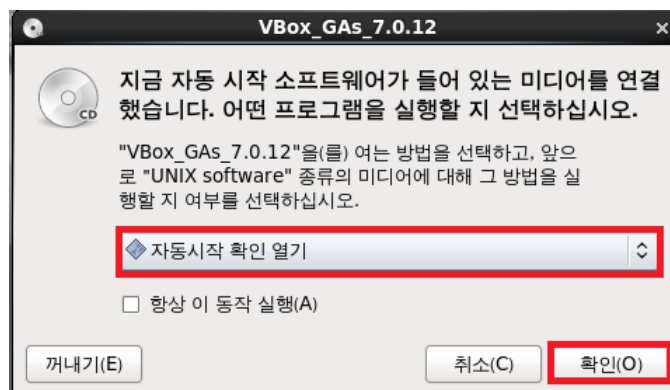
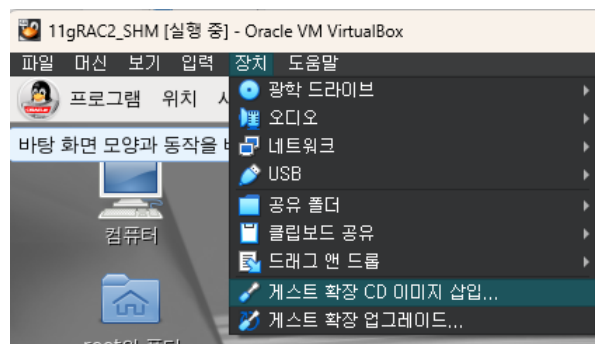
## [ 2번 노드 VM 네트워크 설정 변경 ]

2번 노드를 root로 로그인하여 접속합니다.

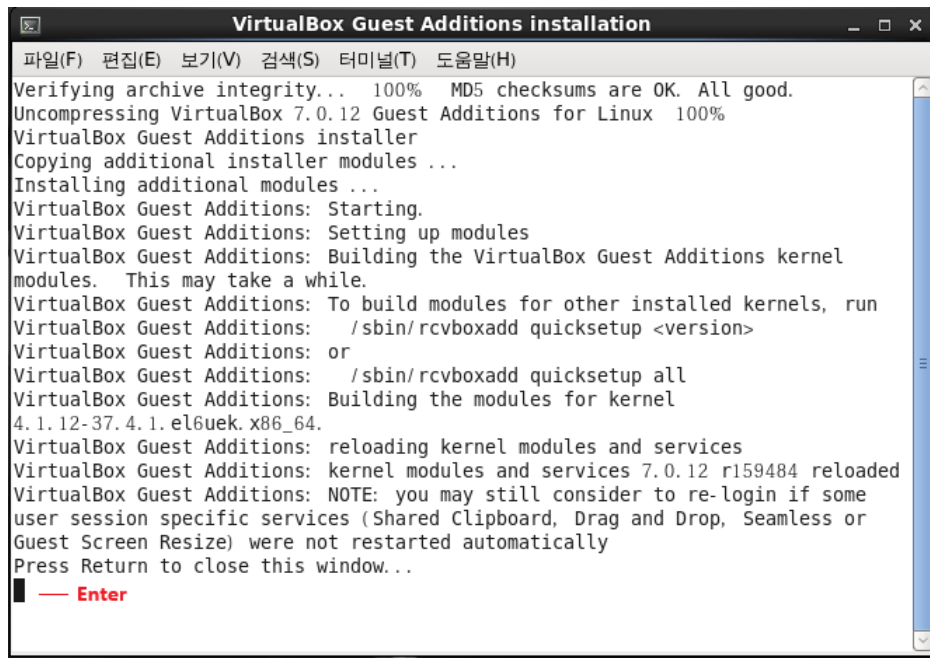




1. 편한한 설정을 위해서 2번 노드 게스트 확장을 수행합니다.

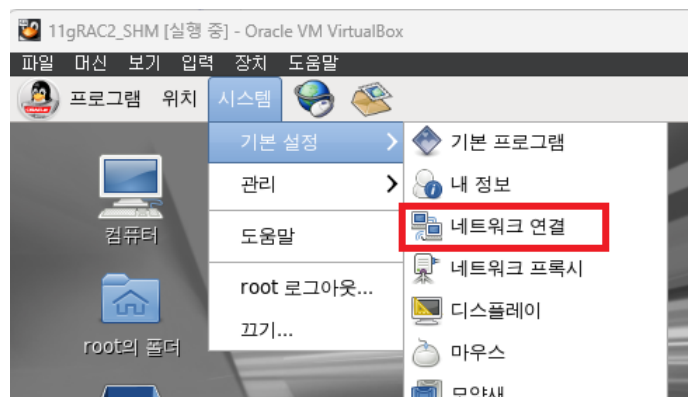


멈추면 enter 누르기

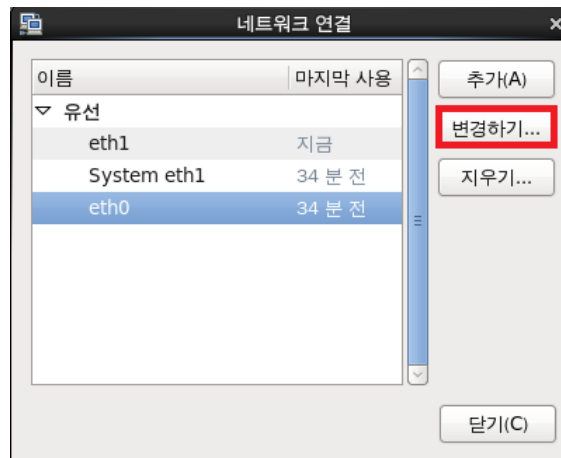


2. 루트 유저로 접속하여 네트워크 환경을 2번 노드에 맞게 변경해야 합니다.

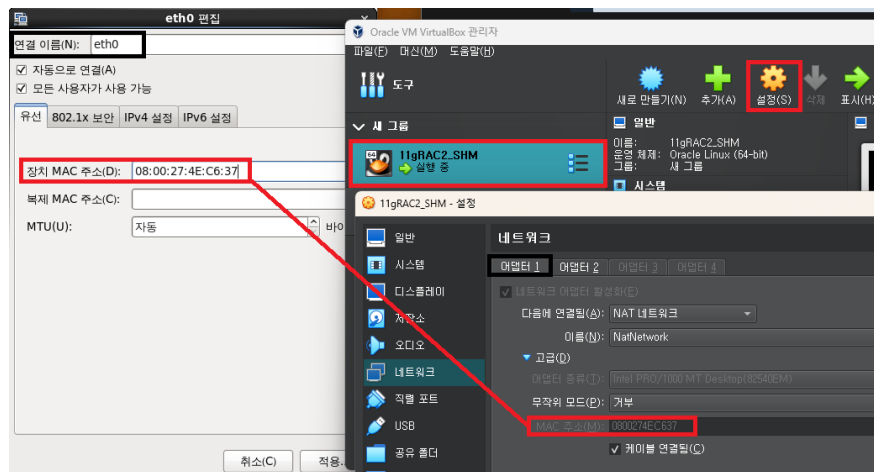
네트워크 IP를 변경하기 위해 '시스템' > '기본설정' > '네트워크'를 선택합니다.



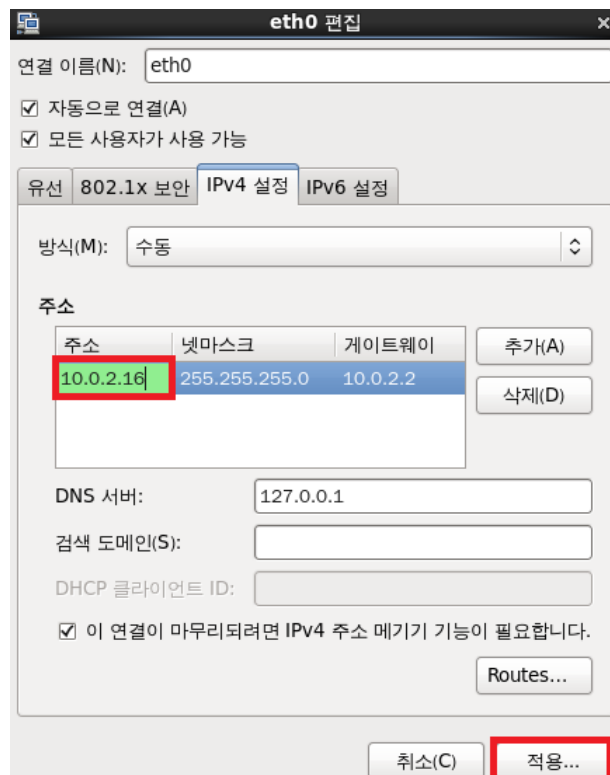
- 2번 노드의 eth0 선택 > 변경하기



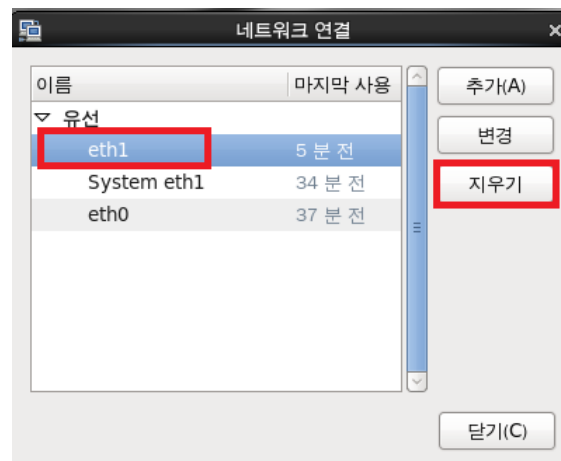
- eth0의 MAC 주소 변경하기 ( 어댑터 1의 MAC 주소 입력 )



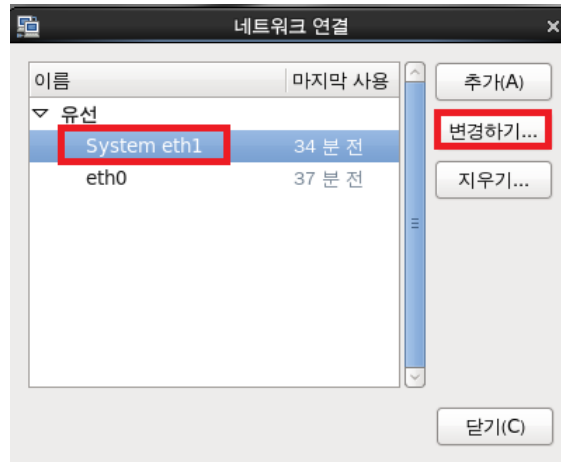
- eth0 IPv4 - 주소: 10.0.2.16 으로 변경 > 적용



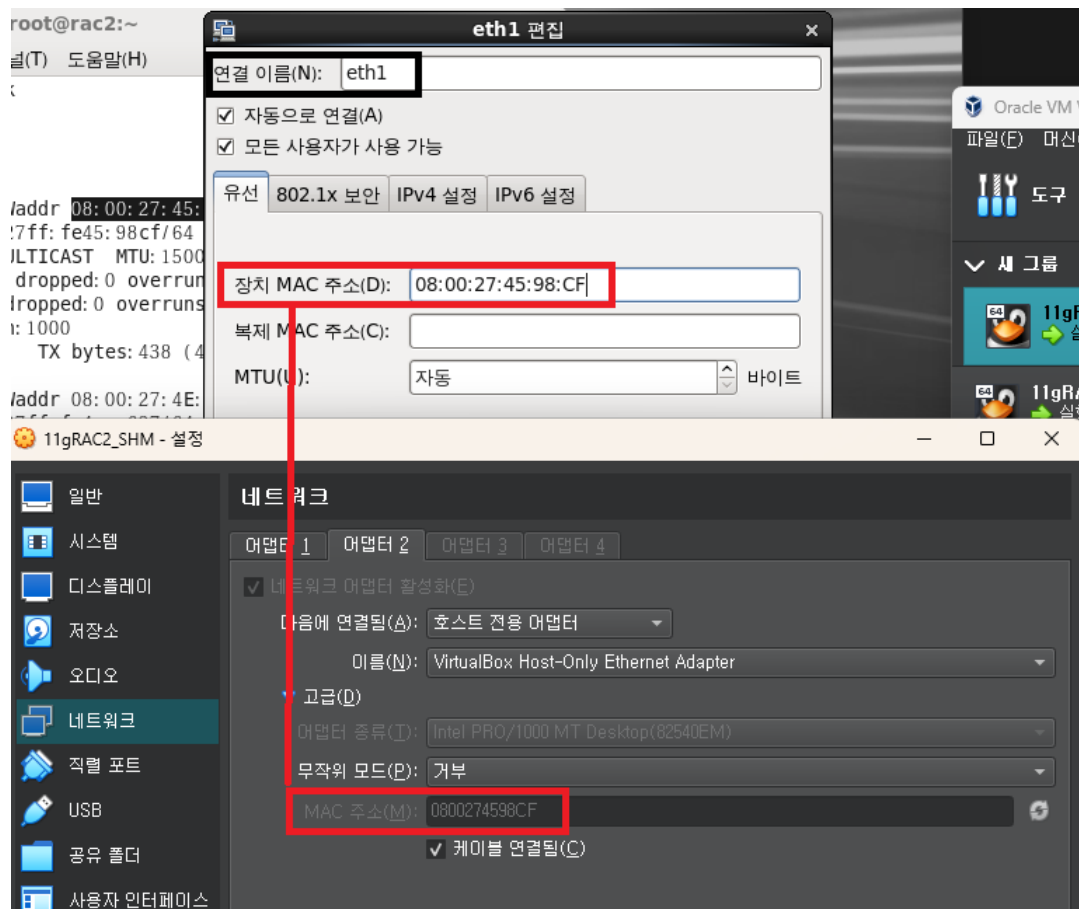
- eth1 선택 > 지우기



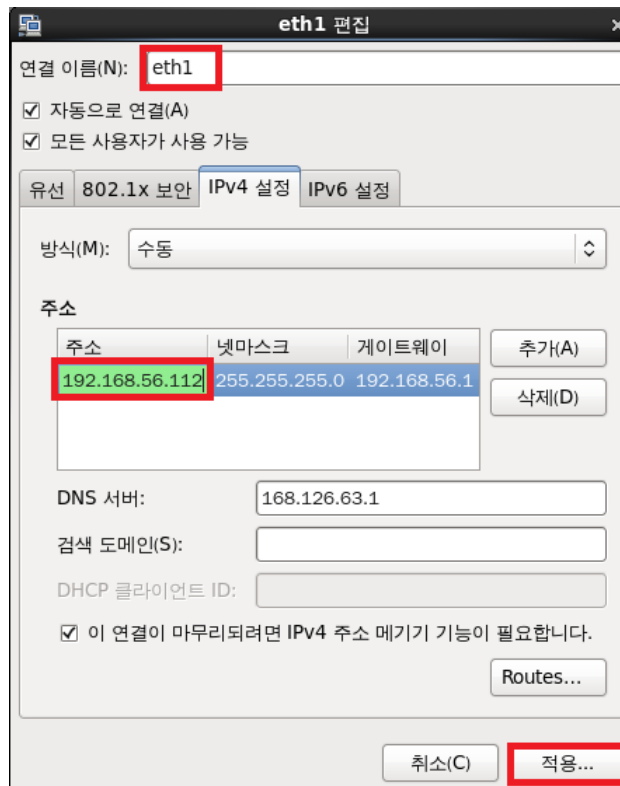
- System eth1 선택 > 변경



- Mac 주소를 변경합니다. ( 어댑터2 의 MAC 주소 입력 )



- eth1으로 이름 변경 후 IPv4 주소 변경 : 192.168.56.112 > 적용



- ifconfig로 주소가 제대로 설정되었는지 확인합니다.

```

root@rac1:~
파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)
eth1 인터페이스 활성화중입니다: 오류: 연결 활성화가 실패했습니다: The connection is not for this device.
[ 실패 ]

[root@rac1 ~]#
[root@rac1 ~]# ifconfig
eth2      Link encap: Ethernet  HWaddr 08:00:27:45:98:CF
          inet6 addr: fe80::a00:27ff:fe45:98cf/64 Scope: Link
          UP BROADCAST RUNNING MULTICAST  MTU: 1500  Metric: 1
          RX packets: 38 errors: 0 dropped: 0 overruns: 0 frame: 0
          TX packets: 7 errors: 0 dropped: 0 overruns: 0 carrier: 0
          collisions: 0 txqueuelen: 1000
          RX bytes: 3240 (3.1 KiB)  TX bytes: 558 (558.0 b)

eth3      Link encap: Ethernet  HWaddr 08:00:27:4E:C6:37
          inet6 addr: fe80::a00:27ff:fe4e:c637/64 Scope: Link
          UP BROADCAST RUNNING MULTICAST  MTU: 1500  Metric: 1
          RX packets: 11 errors: 0 dropped: 0 overruns: 0 frame: 0
          TX packets: 94 errors: 0 dropped: 0 overruns: 0 carrier: 0
          collisions: 0 txqueuelen: 1000
          RX bytes: 660 (660.0 b)  TX bytes: 5778 (5.6 KiB)

lo        Link encap: Local Loopback
          inet addr: 127.0.0.1  Mask: 255.0.0.0
          inet6 addr: ::1/128 Scope: Host
          UP LOOPBACK RUNNING  MTU: 65536  Metric: 1
          RX packets: 104 errors: 0 dropped: 0 overruns: 0 frame: 0
          TX packets: 104 errors: 0 dropped: 0 overruns: 0 carrier: 0
          collisions: 0 txqueuelen: 0
          RX bytes: 7728 (7.5 KiB)  TX bytes: 7728 (7.5 KiB)

[root@rac1 ~]#

```

3. Hostname을 RAC2로 변경합니다.

a. eth0과 eth1 대신 eth2와 eth3이 출력되는 문제를 해결하기 위해서 network를 수정합니다.

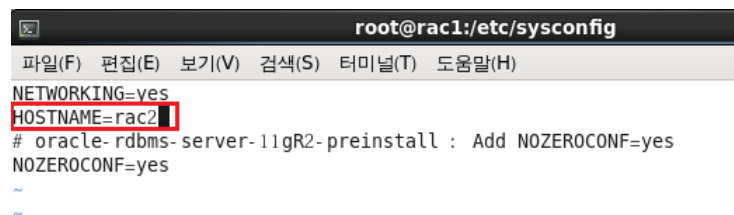
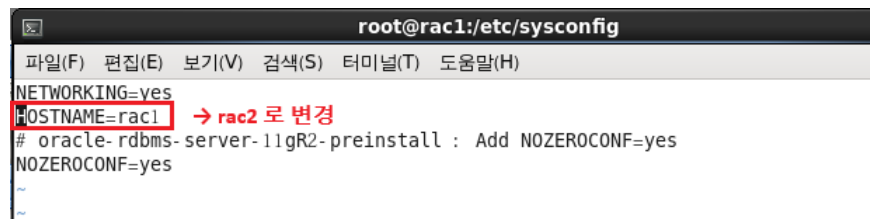
- root 에서 수행

```
# cd /etc/sysconfig
# vi network

----- HOSTNAME을 rac2로 변경 -----
HOSTNAME=rac2
-----
```

▼ 수행해보기

```
| root@rac1 ~]# cd /etc/sysconfig/
| root@rac1 sysconfig]# vi network
```



4. 변경을 완료하였으면, **reboot**을 수행하고 root로 접속하여 ip가 제대로 설정되었는지 확인합니다.

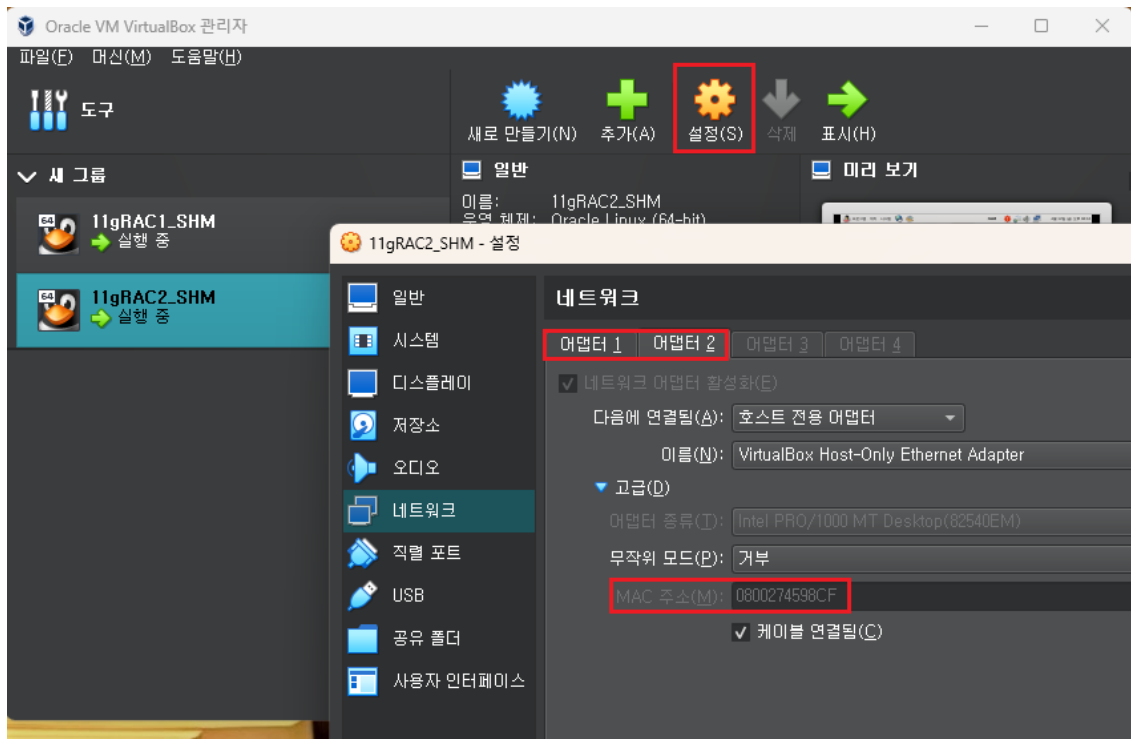
▼ 그래도 안될 경우 조치 사항

`/etc/udev/rules.d/70-persistent-net.rules` 을 열어서 현재 사용하고 있는 MAC주소를 제외한 것들은 삭제합니다.

```
# vi /etc/udev/rules.d/70-persistent-net.rules
```

- 현재 사용 중인 MAC 주소를 확인하는 방법?

VM ware의 11gRAC2\_SHM 설정 > 네트워크 > 어댑터1과 어댑터2의 MAC주소를 확인하면 됩니다. (어댑터1= eth0의 MAC 주소, 어댑터2=eth1의 MAC주소)



1) 70-persistent-net.rules 파일 내용 수정 후 저장

ATTR{address} == "이 주소가 현재 노드들의 MAC주소가 아닌 것은 삭제해줍니다."

```

root@RAC1:/etc/sysconfig/network-scripts
# This file was automatically generated by the /lib/udev/write_net_rules
# program, run by the persistent-net-generator.rules rules file.
#
# You can modify it, as long as you keep each rule on a single
# line, and change only the value of the NAME= key.

# PCI device 0x8086:0x100e (e1000)
SUBSYSTEM=="net", ACTION=="add", DRIVERS=="?*", ATTR{address}=="08:00:27:38:5e:b", ATTR{type}=="1", KERNEL=="eth*", NAME="eth0"
# PCI device 0x8086:0x100e (e1000)
SUBSYSTEM=="net", ACTION=="add", DRIVERS=="?*", ATTR{address}=="08:00:27:09:20:45", ATTR{type}=="1", KERNEL=="eth*", NAME="eth1"
# PCI device 0x8086:0x100e (e1000)
SUBSYSTEM=="net", ACTION=="add", DRIVERS=="?*", ATTR{address}=="08:00:27:e9:52:a", ATTR{type}=="1", KERNEL=="eth*", NAME="eth2"
# PCI device 0x8086:0x100e (e1000)
SUBSYSTEM=="net", ACTION=="add", DRIVERS=="?*", ATTR{address}=="08:00:27:9a:3a:6", ATTR{type}=="1", KERNEL=="eth*", NAME="eth3"


```



(어댑터1= eth0의 MAC 주소, 어댑터2=eth1의 MAC주소)

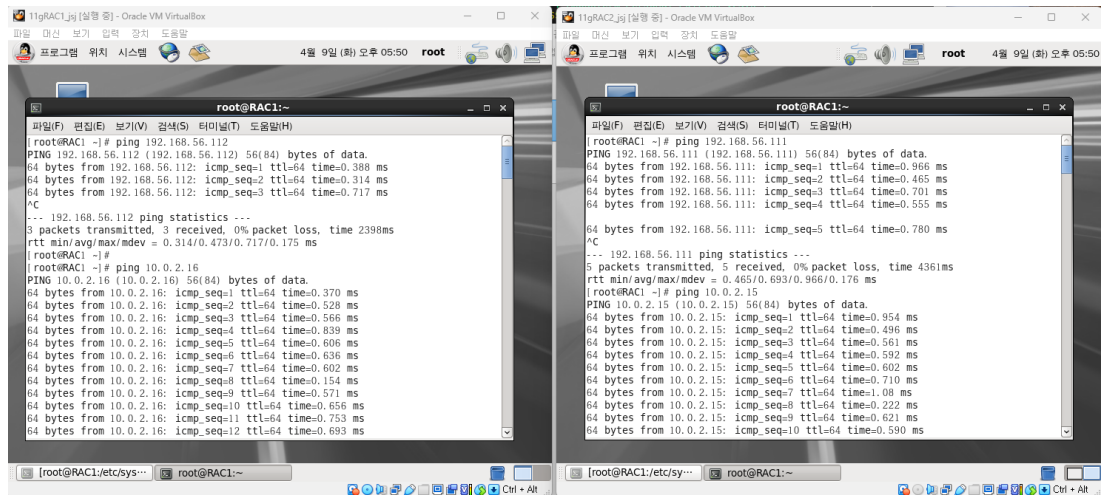
```
root@RAC2:~  
파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)  
# This file was automatically generated by the /lib/udev/write_net_rules  
# program, run by the persistent-net-generator.rules rules file.  
#  
# You can modify it, as long as you keep each rule on a single  
# line, and change only the value of the NAME= key.  
  
# PCI device 0x8086:0x100e (e1000) (custom name provided by external tool)  
SUBSYSTEM=="net", ACTION=="add", DRIVERS=="?*", ATTR{address}=="08:00:27:4e:c6:37",  
ATTR{type}=="l", KERNEL=="eth*", NAME="eth0"  
  
# PCI device 0x8086:0x100e (e1000) (custom name provided by external tool)  
SUBSYSTEM=="net", ACTION=="add", DRIVERS=="?*", ATTR{address}=="08:00:27:45:98:cf",  
ATTR{type}=="l", KERNEL=="eth*", NAME="eth1"  
  
...  
...  
...  
...  
"/etc/udev/rules.d/70-persistent-net.rules" 11L, 665C
```

- 2) reboot
- 3) ifconfig



```
root@rac1:~  
ifconfig  
eth0 Link encap: Ethernet HWaddr 08:00:27:E9:52:AD  
      inet addr: 10.0.2.16 Bcast: 10.0.2.255 Mask: 255.255.255.0  
      inet6 addr: fe80::a00:27ff:fee9:52ad/64 Scope: Link  
      UP BROADCAST RUNNING MULTICAST MTU: 1500 Metric: 1  
      RX packets: 0 errors: 0 dropped: 0 overruns: 0 frame: 0  
      TX packets: 18 errors: 0 dropped: 0 overruns: 0 carrier: 0  
      collisions: 0 txqueuelen: 1000  
      RX bytes: 0 (0.0 b) TX bytes: 1308 (1.2 KiB)  
  
eth1 Link encap: Ethernet HWaddr 08:00:27:9A:3A:6A  
      inet addr: 192.168.56.112 Bcast: 192.168.56.255 Mask: 255.255.255.0  
      inet6 addr: fe80::a00:27ff:fe9a:3a6a/64 Scope: Link  
      UP BROADCAST RUNNING MULTICAST MTU: 1500 Metric: 1  
      RX packets: 0 errors: 0 dropped: 0 overruns: 0 frame: 0  
      TX packets: 34 errors: 0 dropped: 0 overruns: 0 carrier: 0  
      collisions: 0 txqueuelen: 1000  
      RX bytes: 0 (0.0 b) TX bytes: 2268 (2.2 KiB)  
  
lo Link encap: Local Loopback  
   inet addr: 127.0.0.1 Mask: 255.0.0.0  
   inet6 addr: ::1/128 Scope: Host  
   UP LOOPBACK RUNNING MTU: 65536 Metric: 1  
   RX packets: 32 errors: 0 dropped: 0 overruns: 0 frame: 0
```

#### 4) 1&2 노드에서 서로 ping 날리기



- `ifconfig` 로 네트워크 설정을 확인합니다.
  - eth0과 eth1이 잘 뜨고, 주소가 맞게 적용되었는지 확인합니다.

```

root@RAC2:/etc/sysconfig
파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)

[root@RAC2 ~] # ifconfig
eth0      Link encap: Ethernet  HWaddr 08:00:27:4E:C6:37
          inet addr: 10.0.2.16  Bcast: 10.0.2.255  Mask: 255.255.255.0
          inet6 addr: fe80::a00:27ff:fe4e:c637/64 Scope: Link
          UP BROADCAST RUNNING MULTICAST  MTU: 1500  Metric: 1
          RX packets: 7 errors: 0 dropped: 0 overruns: 0 frame: 0
          TX packets: 25 errors: 0 dropped: 0 overruns: 0 carrier: 0
          collisions: 0 txqueuelen: 1000
          RX bytes: 908 (908.0 b)  TX bytes: 1786 (1.7 KiB)

eth1      Link encap: Ethernet  HWaddr 08:00:27:45:98:CF
          inet addr: 192.168.56.112  Bcast: 192.168.56.255  Mask: 255.255.255.0
          inet6 addr: fe80::a00:27ff:fe45:98cf/64 Scope: Link
          UP BROADCAST RUNNING MULTICAST  MTU: 1500  Metric: 1
          RX packets: 0 errors: 0 dropped: 0 overruns: 0 frame: 0
          TX packets: 18 errors: 0 dropped: 0 overruns: 0 carrier: 0
          collisions: 0 txqueuelen: 1000
          RX bytes: 0 (0.0 b)  TX bytes: 1308 (1.2 KiB)

lo        Link encap: Local Loopback
          inet addr: 127.0.0.1  Mask: 255.0.0.0
          inet6 addr: ::1/128 Scope: Host
          UP LOOPBACK RUNNING  MTU: 65536  Metric: 1
          RX packets: 28 errors: 0 dropped: 0 overruns: 0 frame: 0
          TX packets: 28 errors: 0 dropped: 0 overruns: 0 carrier: 0
          collisions: 0 txqueuelen: 0
          RX bytes: 2066 (2.0 KiB)  TX bytes: 2066 (2.0 KiB)
    
```

- `/etc/sysconfig/network-scripts` 로 이동하여 `ifcfg-eth0`과 `ifcfg-eth1`이 존재하는지 확인하고, ip가 잘 구성되었는지 확인합니다.

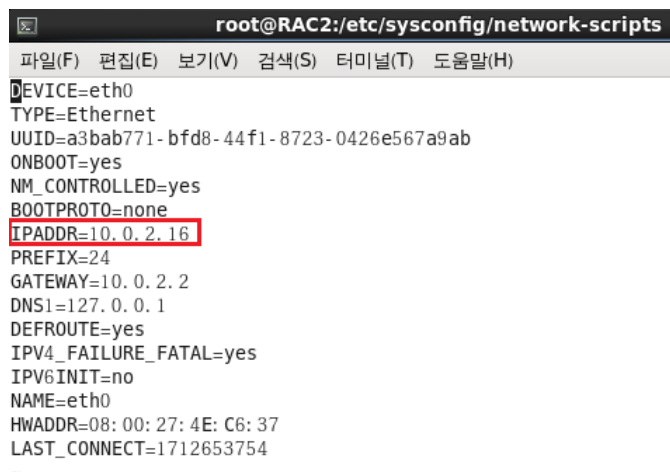
```
# cd /etc/sysconfig/network-scripts
# ls

ifcfg-eth0
ifcfg-eth1
...

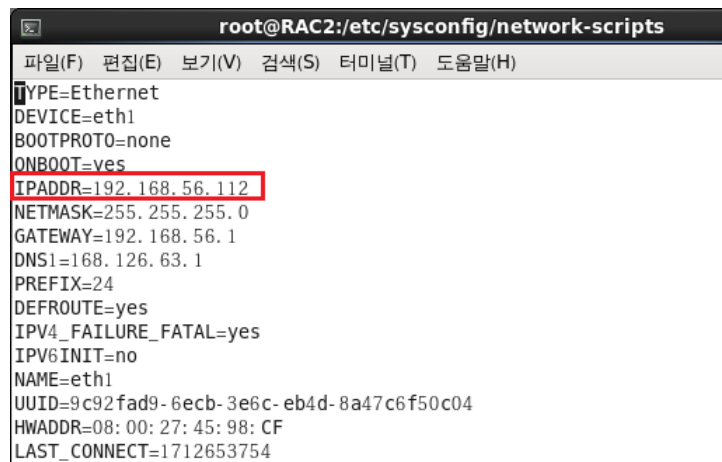
# vi ifcfg-eth0
IPADDR=10.0.2.16

# vi ifcfg-eth1
IPADDR=192.168.56.112
```

▼ 수행 결과



```
root@RAC2:/etc/sysconfig/network-scripts
파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)
DEVICE=eth0
TYPE=Ethernet
UUID=a3bab771-bfd8-44f1-8723-0426e567a9ab
ONBOOT=yes
NM_CONTROLLED=yes
BOOTPROTO=none
IPADDR=10.0.2.16
PREFIX=24
GATEWAY=10.0.2.2
DNS1=127.0.0.1
DEFROUTE=yes
IPV4_FAILURE_FATAL=yes
IPV6INIT=no
NAME=eth0
HWADDR=08:00:27:4E:C6:37
LAST_CONNECT=1712653754
```



```
root@RAC2:/etc/sysconfig/network-scripts
파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)
TYPE=Ethernet
DEVICE=eth1
BOOTPROTO=none
ONBOOT=yes
IPADDR=192.168.56.112
NETMASK=255.255.255.0
GATEWAY=192.168.56.1
DNS1=168.126.63.1
PREFIX=24
DEFROUTE=yes
IPV4_FAILURE_FATAL=yes
IPV6INIT=no
NAME=eth1
UUID=9c92fad9-6ecb-3e6c-eb4d-8a47c6f50c04
HWADDR=08:00:27:45:98:CF
LAST_CONNECT=1712653754
```

5. RAC2에서 rac1으로 ping을 보내서 연결을 확인합니다.

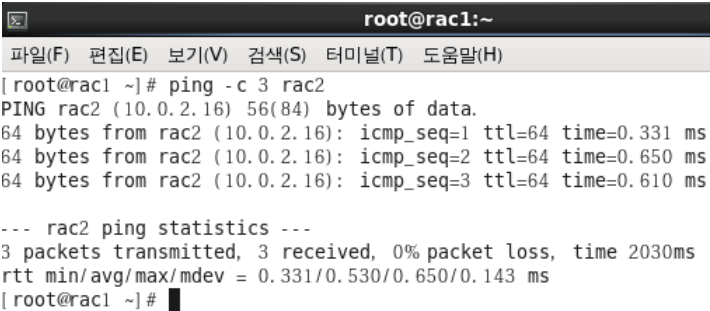
```
# ping -c 3 rac1
```

```
[root@rac2 ~]# ping -c 3 rac1
PING rac1 (10.0.2.15) 56(84) bytes of data.
64 bytes from rac1 (10.0.2.15): icmp_seq=1 ttl=64 time=0.837 ms
64 bytes from rac1 (10.0.2.15): icmp_seq=2 ttl=64 time=0.924 ms
64 bytes from rac1 (10.0.2.15): icmp_seq=3 ttl=64 time=0.556 ms

--- rac1 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2003ms
rtt min/avg/max/mdev = 0.556/0.772/0.924/0.158 ms
```

6. rac1에서 rac2로 ping을 보내서 연결을 확인합니다.

```
# ping -c 3 rac2
```

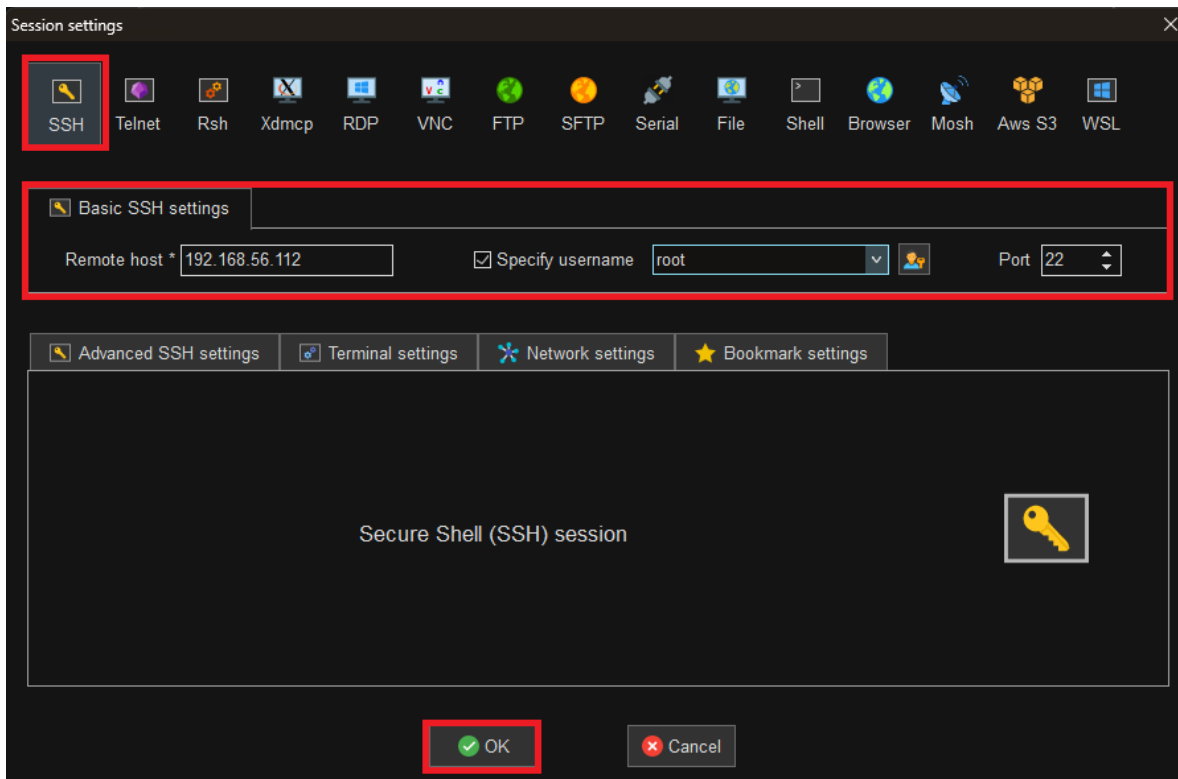


```
root@rac1:~
파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)
[root@rac1 ~]# ping -c 3 rac2
PING rac2 (10.0.2.16) 56(84) bytes of data.
64 bytes from rac2 (10.0.2.16): icmp_seq=1 ttl=64 time=0.331 ms
64 bytes from rac2 (10.0.2.16): icmp_seq=2 ttl=64 time=0.650 ms
64 bytes from rac2 (10.0.2.16): icmp_seq=3 ttl=64 time=0.610 ms

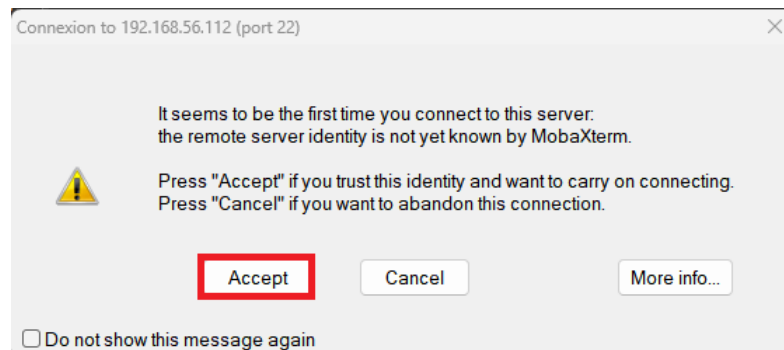
--- rac2 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2030ms
rtt min/avg/max/mdev = 0.331/0.530/0.650/0.143 ms
[root@rac1 ~]#
```

## ▼ rac2 모바텀 연결/사용자 환경 파일 수정/리소스 절약 모드로 변경

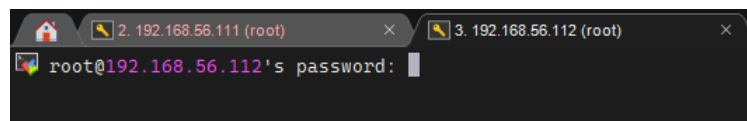
모바텀을 열어서 192.168.56.112 세션을 설정합니다.



아래와 같이 경고 창이 뜨면 Accept 선택



비밀번호를 입력합니다. (oracle)



oracle 유저로 접속하여 .bash\_profile을 수정합니다.

- rac1으로 작성되어 있는 부분을 2로 변경

```
[root@RAC2 ~]# su - oracle
[oracle@racdb1 ~]$
[oracle@racdb1 ~]$ vi .bash_profile
```

```

----- 아래의 부분 수정 -----
export ORACLE_HOSTNAME=rac2      # rac2 => 2번 노드
export ORACLE_SID=racdb2        # racdb2 => 2번 노드
-----

[oracle@racdb1 ~]$ source .bash_profile # 변경 사항 적용
[oracle@racdb2 ~]$ # racdb2로 변경 된 것 확인 가능

```

```

[root@rac2 ~]# su - oracle
[oracle@racdb1 ~]$
[oracle@racdb1 ~]$ vi .bash_profile
[oracle@racdb1 ~]$
[oracle@racdb1 ~]$ source .bash_profile
[oracle@racdb2 ~]$

```

```

# .bash_profile

# Get the aliases and functions
if [ -f ~/.bashrc ]; then
    . ~/.bashrc
fi

# User specific environment and startup programs

PATH=$PATH:$HOME/bin

export PATH
export TMP=/tmp
export TMPDIR=$TMP
export EDITOR=vi
export ORACLE_HOSTNAME=rac2      # rac2 => 2번 노드
export ORACLE_UNQNAME=racdb
export ORACLE_BASE=/u01/app/oracle
export GRID_HOME=/u01/app/11.2.0/grid
export DB_HOME=$ORACLE_BASE/product/11.2.0/db_1
export ORACLE_HOME=$DB_HOME
export ORACLE_SID=racdb2        # racdb2 => 2번 노드
export ORACLE_TERM=xterm
export BASE_PATH=/usr/sbin:$PATH
export PATH=$ORACLE_HOME/bin:$ORACLE_HOME/OPatch:$GRID_HOME/bin:$BASE_PATH
export LD_LIBRARY_PATH=$ORACLE_HOME/lib:/lib:/usr/lib
export CLASSPATH=$ORACLE_HOME/JRE:$ORACLE_HOME/jlib:$ORACLE_HOME/rdbms/jlib
export NLS_LANG=AMERICAN_KOREA.AL32UTF8
export PS1='[\u@$ORACLE_SID \w]$ '
~

```

grid 유저로 접속하여 .bash\_profile에 ra1으로 설정된 이름을 변경합니다.

```

[oracle@racdb2 ~]$ su - grid
암호: _____ #암호 : oracle
[grid@+ASM1 ~]$ vi .bash_profile

----- 아래의 내용 수정 -----
export ORACLE_HOSTNAME=rac2      # rac2 => 2번 노드
export ORACLE_SID=+ASM2        # +ASM2 => 2번 노드
-----

[grid@+ASM1 ~]$ source .bash_profile
[grid@+ASM2 ~]$

```

```
[oracle@racdb2 ~]$ su - grid
암호 :
[grid@ASM1 ~]$ vi .bash_profile
[grid@ASM1 ~]$
[grid@ASM1 ~]$ source .bash_profile
[grid@ASM2 ~]$
```

```
# .bash_profile

# Get the aliases and functions
if [ -f ~/.bashrc ]; then
    . ~/.bashrc
fi

# User specific environment and startup programs

PATH=$PATH:$HOME/bin

export PATH
export TMP=/tmp
export TMPDIR=$TMP
export EDITOR=vi
export ORACLE_HOSTNAME=rac2 # rac2 => 2번 노드
export ORACLE_BASE=/u01/app/oracle
export GRID_HOME=/u01/app/11.2.0/grid
export DB_HOME=$ORACLE_BASE/product/11.2.0/db_1
export ORACLE_HOME=$GRID_HOME
export ORACLE_SID=+ASM2 # +ASM2 => 2번 노드
export ORACLE_TERM=xterm
export BASE_PATH=/usr/sbin:$PATH
export PATH=$ORACLE_HOME/bin:$ORACLE_HOME/OPatch:$BASE_PATH
export LD_LIBRARY_PATH=$ORACLE_HOME/lib:/lib:/usr/lib
export CLASSPATH=$ORACLE_HOME/JRE:$ORACLE_HOME/jlib:$ORACLE_HOME/rdbms/jlib
export NLS_LANG=AMERICAN_AMERICA.AL32UTF8
export PS1='\u@$ORACLE_SID \w]$ '
```

## 선택사항

리소스 절약을 위해 2번 노드는 텍스트 모드로 가동하고자 합니다.

root 유저 권한으로 /etc/inittab을 수정해서 텍스트 모드로 부팅되도록 변경합니다.

### • vi /etc/inittab

id:3:initdefault: ##### 5를 3으로 변경

```
root@rac2:~
파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)
# For information on how to write upstart event handlers, or how
# upstart works, see init(5), init(8), and initctl(8).
#
# Default runlevel. The runlevels used are:
# 0 - halt (Do NOT set initdefault to this)
# 1 - Single user mode
# 2 - Multiuser, without NFS (The same as 3, if you do not have networking)
# 3 - Full multiuser mode
# 4 - unused
# 5 - X11
# 6 - reboot (Do NOT set initdefault to this)
#
id:5:initdefault: → id:3:initdefault 로 변경하기
```

```

root@rac2:~
파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)
# For information on how to write upstart event handlers, or how
# upstart works, see init(5), init(8), and initctl(8).
#
# Default runlevel. The runlevels used are:
# 0 - halt (Do NOT set initdefault to this)
# 1 - Single user mode
# 2 - Multiuser, without NFS (The same as 3, if you do not have networking)
# 3 - Full multiuser mode
# 4 - unused
# 5 - X11
# 6 - reboot (Do NOT set initdefault to this)
#
id:3:initdefault:
-- INSERT --

```

정지한 1번 노드를 부팅하고, 2번 노드를 재기동 한 후에 양쪽 노드간의 정상적인 통신이 가능한지 확인합니다.

- 텍스트 모드로 변경하여 rac2 서버 화면이 아래와 같이 보입니다.( root로 로그인 pw: oracle )

```

11gRAC2_SHM [실행 중] - Oracle VM VirtualBox
파일  머신  보기  입력  장치  도움말
Oracle Linux Server release 6.8
Kernel 4.1.12-37.4.1.el6uek.x86_64 on an x86_64
RAC2 login:

```

```

11gRAC2_SHM [실행 중] - Oracle VM VirtualBox
파일  머신  보기  입력  장치  도움말
Oracle Linux Server release 6.8
Kernel 4.1.12-37.4.1.el6uek.x86_64 on an x86_64
RAC2 login: root
Password:

```

```

11gRAC2_SHM [실행 중] - Oracle VM VirtualBox
파일  머신  보기  입력  장치  도움말
Oracle Linux Server release 6.8
Kernel 4.1.12-37.4.1.el6uek.x86_64 on an x86_64
RAC2 login: root
Password:
Last login: Tue Apr  9 19:37:56 from 192.168.56.1
root@rac2 ~]#
root@rac2 ~]#

```

원래대로 돌아가고 싶으면 다시 아래와 같이 수정하고 reboot 하면 됩니다.

- vi /etc/inittab



id:5:initdefault:

### 3을 5로 변경

## 네트워크 검증

- root 유저로 수행

```
ping -c 3 rac1
ping -c 3 rac2
ping -c 3 rac1-priv
ping -c 3 rac2-priv
nslookup rac-scan
```

### ▼ 실행해보기

```
[root@rac1 ~]# ping -c 3 rac1
PING rac1.localdomain (10.0.1.101) 56(84) bytes of data.
64 bytes from rac1.localdomain (10.0.1.101): icmp_seq=1 ttl=64 time=0.013 ms
64 bytes from rac1.localdomain (10.0.1.101): icmp_seq=2 ttl=64 time=0.029 ms
64 bytes from rac1.localdomain (10.0.1.101): icmp_seq=3 ttl=64 time=0.017 ms

--- rac1.localdomain ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2000ms
rtt min/avg/max/mdev = 0.013/0.019/0.029/0.008 ms

[root@rac1 ~]# ping -c 3 rac2
PING rac2.localdomain (10.0.1.102) 56(84) bytes of data.
64 bytes from rac2.localdomain (10.0.1.102): icmp_seq=1 ttl=64 time=0.659 ms
64 bytes from rac2.localdomain (10.0.1.102): icmp_seq=2 ttl=64 time=0.298 ms
64 bytes from rac2.localdomain (10.0.1.102): icmp_seq=3 ttl=64 time=0.743 ms

--- rac2.localdomain ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2000ms
rtt min/avg/max/mdev = 0.298/0.566/0.743/0.194 ms

[root@rac1 ~]# ping -c 3 rac1-priv
PING rac1-priv.localdomain (10.0.5.101) 56(84) bytes of data.
64 bytes from rac1-priv.localdomain (10.0.5.101): icmp_seq=1 ttl=64 time=0.016 m
s
64 bytes from rac1-priv.localdomain (10.0.5.101): icmp_seq=2 ttl=64 time=0.038 m
s
64 bytes from rac1-priv.localdomain (10.0.5.101): icmp_seq=3 ttl=64 time=0.034 m
s

--- rac1-priv.localdomain ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2002ms
rtt min/avg/max/mdev = 0.016/0.029/0.038/0.010 ms

[root@rac1 ~]# ping -c 3 rac2-priv
PING rac2-priv.localdomain (10.0.5.102) 56(84) bytes of data.
64 bytes from rac2-priv.localdomain (10.0.5.102): icmp_seq=1 ttl=64 time=0.698 m
```

```
S
64 bytes from rac2-priv.localdomain (10.0.5.102): icmp_seq=2 ttl=64 time=1.02 ms
64 bytes from rac2-priv.localdomain (10.0.5.102): icmp_seq=3 ttl=64 time=0.851 m
S

--- rac2-priv.localdomain ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2002ms
rtt min/avg/max/mdev = 0.698/0.859/1.029/0.137 ms

[root@rac1 ~]# nslookup rac-scan
Server:          127.0.0.1
Address:         127.0.0.1#53

Name:   rac-scan
Address: 10.0.1.110
```